



Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Estado de México

TC3006C. Inteligencia artificial avanzada para la ciencia de datos

Grupo: 101

Iván Alexander Ramos Ramírez - A01750817

Fecha de entrega: 14 de Septiembre del 2025

## Descripción del dataset Digits:

El dataset Digits forma parte de la librería scikit-learn y corresponde a un conjunto de datos clásico para tareas de clasificación supervisada. Este conjunto contiene 1797 muestras, cada una representando un dígito manuscrito en el rango 0 a 9, lo que constituye un problema de clasificación multiclase con 10 clases distintas.

Cada imagen se encuentra en formato de matriz de  $8 \times 8$  píxeles, con valores enteros entre 0 y 16 que indican la intensidad en escala de grises. Para facilitar su uso en algoritmos de aprendizaje automático, cada imagen es aplanada en un vector de 64 características numéricas, lo que permite representar los píxeles como un conjunto de variables de entrada.



Figura 1 ejemplo de los labels

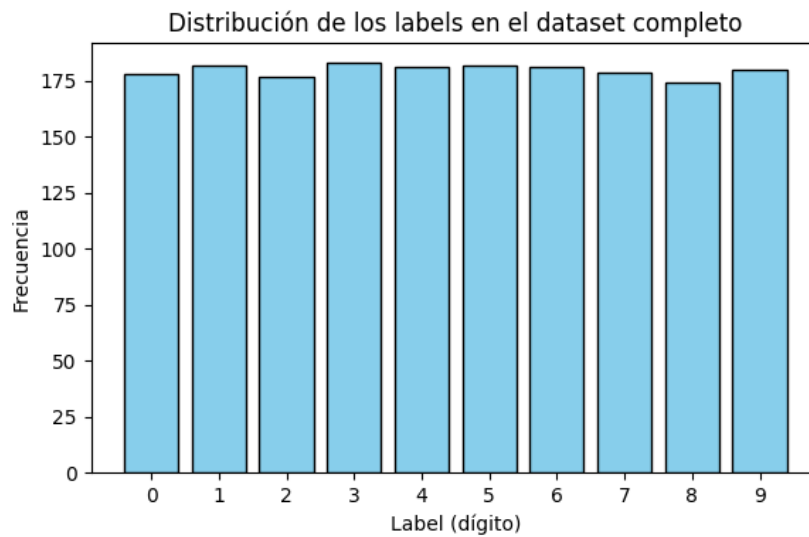
La variable objetivo (target) corresponde al valor numérico real del dígito representado en cada imagen. Dado que las clases se encuentran relativamente balanceadas, el dataset es apropiado para la evaluación de modelos de clasificación y el análisis de técnicas de regularización y ajuste de hiperparámetros.

### Target:

El análisis porcentual de la distribución de los labels muestra que todas las clases tienen una proporción muy similar, alrededor del 10%. Esto indica que el dataset está muy balanceado y no existe predominancia de ninguna clase sobre las demás. Un conjunto de datos balanceado es ideal para el entrenamiento de modelos de clasificación, ya que permite que el modelo aprenda a reconocer todas las clases por igual y evita problemas de sesgo hacia las clases más frecuentes. Por lo tanto, los resultados y métricas obtenidas reflejan de manera justa el desempeño del modelo en todas las clases.

```
Label 0: 9.91%
Label 1: 10.13%
Label 2: 9.85%
Label 3: 10.18%
Label 4: 10.07%
Label 5: 10.13%
Label 6: 10.07%
Label 7: 9.96%
Label 8: 9.68%
Label 9: 10.02%
```

Esto se puede visualizar de mejor manera en el siguiente grafico de distribución de los targets del dataset.



### Separación de los conjuntos:

Se realizó una división estratificada de los datos en tres subconjuntos:

```
original Shape: (1797, 64) (1797,)
Train shape: (1265, 64) (1265,) 70.4%
Test shape: (266, 64) (266,) 14.8%
Validation shape: (266, 64) (266,) 14.8%
```

- 70% entrenamiento
- 15% validación
- 15% prueba.

La separación estratificada asegura que la distribución de clases se mantenga similar en cada subconjunto, evitando sesgos y permitiendo una evaluación más confiable. El conjunto de validación es esencial para ajustar hiperparámetros de manera objetiva, ya que, si se usara el conjunto de prueba para este fin, se correría el riesgo de sobreajustar el modelo a los datos de prueba y obtener métricas poco representativas.

### **Preprocesamiento: Escalado de características**

Antes de entrenar la red neuronal, se realizó un escalado de las variables de entrada (features) utilizando la media y desviación estándar calculadas únicamente sobre el conjunto de entrenamiento. Este procedimiento transforma cada píxel para que tenga media cercana a cero y desviación estándar igual a uno.

El motivo principal para escalar las características es mejorar la estabilidad y eficiencia del proceso de aprendizaje. Las redes neuronales son sensibles a la magnitud de las entradas: si los valores de los píxeles varían en rangos muy distintos, las actualizaciones de los pesos pueden volverse inestables y el modelo puede tardar más en converger o incluso no aprender correctamente. Además, el escalado ayuda a que todas las características tengan una influencia comparable en el entrenamiento, evitando que aquellas con mayor rango dominen el proceso de optimización.

Por lo tanto, el escalado es una práctica recomendada en el preprocesamiento de datos para modelos de machine learning, especialmente en redes neuronales, y contribuye a obtener mejores resultados y métricas de desempeño.

### **Análisis inicial del desempeño del modelo.**

Accuracy: 0.9774436090225563				
Reporte de clasificación:				
	precision	recall	f1-score	support
0	1.000	1.000	1.000	26
1	0.962	0.926	0.943	27
2	1.000	1.000	1.000	26
3	0.964	1.000	0.982	27
4	0.931	1.000	0.964	27
5	0.963	0.963	0.963	27
6	1.000	0.963	0.981	27
7	1.000	0.962	0.980	26
8	1.000	0.962	0.980	26
9	0.964	1.000	0.982	27
accuracy			0.977	266
macro avg	0.978	0.977	0.978	266
weighted avg	0.978	0.977	0.977	266

- Accuracy: 0.977 (97.7%), muy alto.
- Precision, recall y f1-score: Todos los valores por clase están cerca de 1.0, lo que significa que el modelo identifica correctamente casi todos los dígitos y comete muy pocos errores.
- Macro y weighted avg: También muy altos, lo que confirma que el desempeño es consistente en todas las clases y no hay clases desbalanceadas afectando el resultado.

### Diagnóstico y explicación del grado de bias o sesgo

- Alto bias (subajuste): baja precisión en train y en val/test.
- Medio bias: precisión moderada en train; hay margen de mejora.
- Bajo bias: alta precisión en train; si hay gap grande vs val/test, el problema es varianza.

El modelo no presenta problemas de sesgo ni de sobreajuste. La red neuronal logra aprender los patrones relevantes del dataset y generaliza correctamente a datos no vistos.

```

Train acc=0.999 loss=0.011
Val   acc=0.951 loss=0.279 (gap=0.048)
Test  acc=0.977 loss=0.146 (gap=0.022)
Diagnóstico de bias: bajo
Indicadores: alta precisión en train. Si el gap es grande, el problema es varianza (overfitting)

```

- El sesgo (bias) es bajo: el modelo tiene suficiente capacidad y aprende bien los patrones, por lo que podemos descartar underfitting.
- Este test no muestra evidencia de la presencia de overfitting, la precisión en validación y prueba es alta y cercana a la de entrenamiento, lo que indica buena generalización.

Esto se logra visualizar mejor en la Figura 2

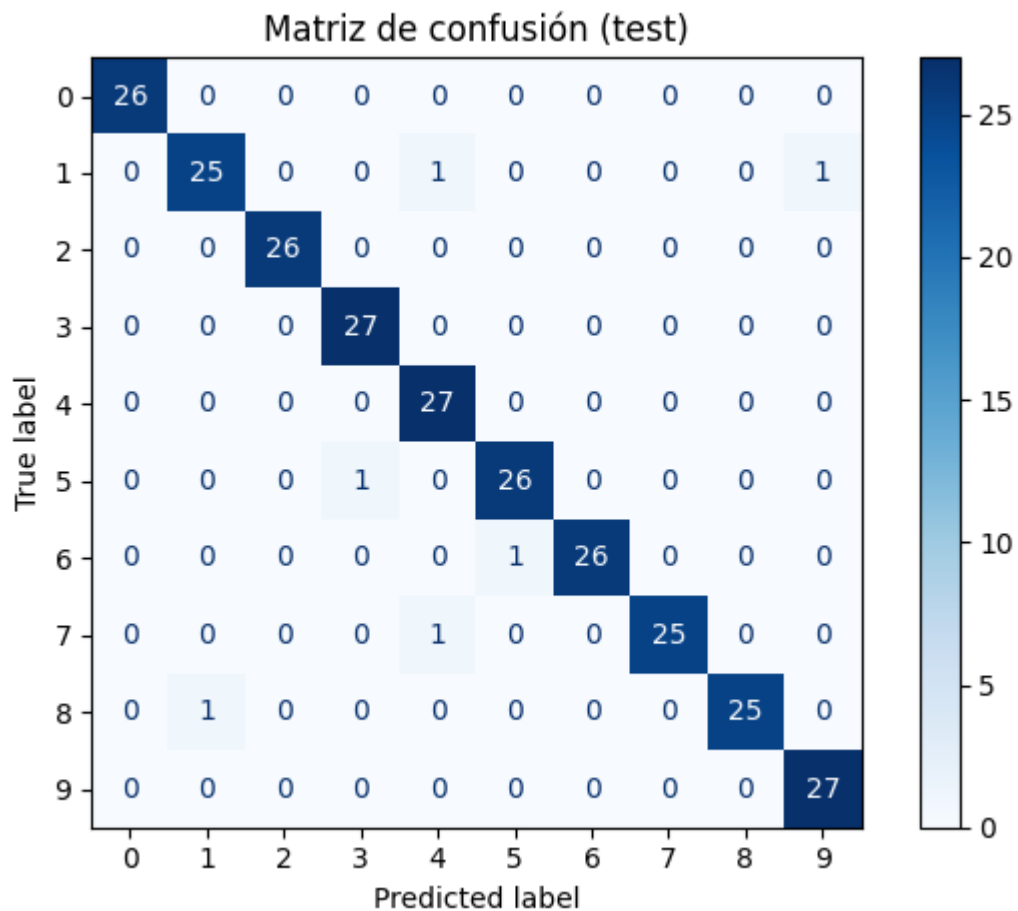


Figura 2 Matriz de confusión.

### Diagnóstico y explicación el grado varianza

Para diagnosticar el grado de varianza del modelo, se compara la precisión obtenida en el conjunto de entrenamiento con la precisión en los conjuntos de validación y prueba. La diferencia entre estas métricas, conocida como “gap”, indica qué tan bien el modelo generaliza a datos no vistos.

En la práctica, se utilizan umbrales estándar para clasificar el gap:

- Gap menor a 0.03: varianza baja (el modelo generaliza bien).
- Gap entre 0.03 y 0.08: varianza media (hay algo de sobreajuste, pero no crítico).
- Gap mayor a 0.08: varianza alta (el modelo sobreajusta los datos de entrenamiento).

Estos rangos permiten interpretar objetivamente el desempeño del modelo y tomar decisiones sobre ajustes en la arquitectura o regularización, facilitando el análisis sin necesidad de recurrir a fuentes externas.

```
Varianza (train vs val): medio (gap=0.048)
Varianza (train vs test): bajo (gap=0.022)
Indicadores: hay algo de sobreajuste, pero no crítico. Puedes ajustar regularización o probar más datos.
```

Lo que nos indica en la prueba que hay una ligera varianza más alta al probar el modelo con el split de validación, lo que indica un ligero overfitting, sin embargo, este se puede manejar en los hiperparametros para tener mejores resultados, esto implementando un algoritmo de regularización para evitar que el modelo aprenda mas generalizadamente.

Podemos ver mejor esta diferencia entre fracciones de ambos conjuntos (validacion y testing) en la figura 3.

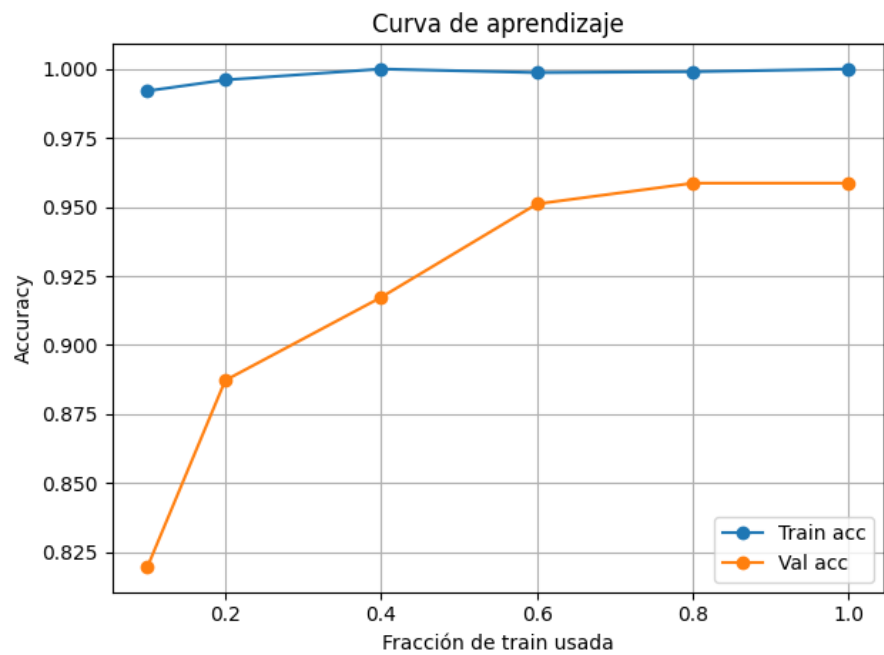


Figura 3



```
Epoch 1 | loss=2.3575 | acc=0.065
Epoch 10 | loss=2.3249 | acc=0.077
Epoch 20 | loss=2.2947 | acc=0.098
Epoch 30 | loss=2.2678 | acc=0.136
Epoch 40 | loss=2.2411 | acc=0.172
Epoch 40 | loss=2.2411 | acc=0.172
Epoch 50 | loss=2.2132 | acc=0.217
Epoch 60 | loss=2.1840 | acc=0.257
Epoch 70 | loss=2.1503 | acc=0.295
Epoch 80 | loss=2.1110 | acc=0.340
Epoch 50 | loss=2.2132 | acc=0.217
Epoch 60 | loss=2.1840 | acc=0.257
Epoch 70 | loss=2.1503 | acc=0.295
Epoch 80 | loss=2.1110 | acc=0.340
Epoch 90 | loss=2.0669 | acc=0.375
Epoch 90 | loss=2.0669 | acc=0.375
Epoch 100 | loss=2.0168 | acc=0.413
Epoch 110 | loss=1.9629 | acc=0.464
Epoch 120 | loss=1.9014 | acc=0.499
Epoch 130 | loss=1.8344 | acc=0.532
Epoch 100 | loss=2.0168 | acc=0.413
Epoch 110 | loss=1.9629 | acc=0.464
Epoch 120 | loss=1.9014 | acc=0.499
Epoch 130 | loss=1.8344 | acc=0.532
Epoch 140 | loss=1.7594 | acc=0.551
...
Epoch 990 | loss=0.0531 | acc=0.994
Epoch 1000 | loss=0.0516 | acc=0.994
Accuracy test: 0.959
Loss test: 0.181
```